

BAB II

LANDASAN TEORI

2.1 Distribusi dan Transportasi

Logistik adalah salah satu bagian dari *Supply Chain Management* yang mengatur alur produk mulai titik awal sampai ke titik tujuan. Tujuan akhir yang ingin dicapai melalui manajemen logistik adalah mendapatkan sejumlah barang atau jasa yang tepat pada tempat dan waktu yang tepat, serta kondisi yang diinginkan dengan memberikan kontribusi terbesar bagi perusahaan (Ballou, 2004).

Dalam melakukan kegiatan perpindahan produk dibutuhkan suatu sistem yang disebut dengan distribusi dan transportasi. Menurut Pujawan (2005), secara umum fungsi distribusi dan transportasi pada dasarnya adalah menghantarkan produk dari lokasi dimana produk tersebut diproduksi sampai dimana mereka akan digunakan. Dalam upaya memenuhi tujuan tersebut manajemen distribusi dan transportasi melakukan sejumlah fungsi dasar yang terdiri dari (Pujawan, 2005):

1. Melakukan segmentasi dan menentukan target *service level*. Segmentasi pelanggan perlu dilakukan karena kontribusi mereka pada *revenue* perusahaan bisa sangat bervariasi dan karakteristik tiap pelanggan bisa berbeda satu dengan yang lain.
2. Menentukan mode transportasi yang digunakan. Tiap mode transportasi memiliki karakteristik yang berbeda dan mempunyai keunggulan serta kelemahan yang berbeda juga. Manajemen transportasi harus bisa menentukan mode apa yang akan digunakan dalam pengiriman atau mendistribusikan produk-produk mereka ke pelanggan.
3. Melakukan konsolidasi informasi dan pengiriman. Konsolidasi merupakan kata kunci yang sangat penting. Tekanan untuk melakukan pengiriman cepat namun murah menjadi pendorong utama perlunya melakukan konsolidasi informasi maupun pengiriman.

4. Melakukan penjadwalan dan penentuan rute pengiriman. Salah satu kegiatan operasional yang dilakukan oleh gudang atau distributor adalah menentukan kapan sebuah truk harus berangkat dan rute mana yang harus dilalui untuk memenuhi permintaan dari sejumlah pelanggan.
5. Memberikan pelayanan nilai tambah. Disamping mengirim produk ke pelanggan jaringan distribusi semakin banyak dipercaya untuk melakukan proses nilai tambah.
6. Menyimpan persediaan. Jaringan distribusi selalu melibatkan proses penyimpanan produk baik di suatu gudang pusat atau gudang regional, maupun di toko di mana produk tersebut di pajang untuk dijual.
7. Manajemen pengembalian. Manajemen distribusi juga bertanggung jawab untuk melaksanakan kegiatan pengembalian produk dari hilir ke hulu dalam rantai pasok.

Pengambilan keputusan pada pengiriman produk merupakan hal yang sangat penting dikarenakan biaya yang dikeluarkan oleh perusahaan pada saat mengirim produk ke pelanggan cukup besar. Menurut Ballou (2004), biaya transportasi berkisar antara sepertiga atau duapertiga dari total biaya logistik sehingga peningkatan efisiensi dengan memaksimalkan utilitas alat dan personil transportasi menjadi permasalahan utama. Salah satu cara mengoptimalkan biaya transportasi dengan menentukan rute pengiriman yang tepat untuk memenuhi semua permintaan dari pelanggan.

2.2 Vehicle Routing Problem

2.2.1 Definisi Vehicle Routing Problem

Vehicle routing problem merupakan permasalahan dalam penentuan rute kendaraan yang diawali dari sebuah depot menuju ke sejumlah konsumen yang memiliki permintaan dengan tujuan untuk meminimasi jarak, waktu dan biaya yang dikeluarkan dan harus sesuai dengan batasan-batasan atau karakteristik yang telah ditentukan. Toth dan Vigo (2002), mendefinisikan VRP sebagai suatu pencarian solusi yang meliputi penentuan sejumlah rute, dimana masing-masing rute dilalui oleh satu kendaraan yang berawal dan berakhir di depot asalnya, sehingga

permintaan semua pelanggan terpenuhi dengan tetap memenuhi kendala operasional yang ada dan juga meminimalisasi biaya transportasi global.

VRP merupakan sebuah permasalahan dalam kategori *non-polynomial hard problem*, yang berarti waktu komputasi yang digunakan akan semakin tinggi secara *non deterministic polynomial* seiring dengan meningkatnya ukuran dari permasalahan. Sejauh ini pendekatan yang dipakai untuk problem VRP adalah pendekatan analitik dan metaheuristik untuk mencari solusi berupa rute yang optimal atau mendekati optimal (Santosa dan Willy, 2011). Terdapat beberapa tujuan VRP yang diungkapkan oleh Toth dan Vigo (2002), diantaranya:

1. Minimasi biaya transportasi keseluruhan, biaya ini dipengaruhi oleh jarak tempuh ataupun waktu tempuh dan biaya tetap yang digunakan dalam penggunaan kendaraan.
2. Minimasi jumlah kendaraan yang digunakan dalam melayani *customer*.
3. Menyeimbangkan rute, waktu tempuh dan kapasitas muatan.
4. Minimasi kerugian ataupun keluhan dari pelanggan.

2.2.2 Jenis-Jenis Vehicle Routing Problem

Dalam penggunaan VRP pada dunia nyata, banyak faktor sampingan yang muncul. Faktor-faktor tersebut berpengaruh terhadap munculnya variasi dari VRP dengan penambahan batasan-batasan atau *constraint*. Menurut Toth dan Vigo (2002), ada beberapa variasi permasalahan dalam VRP, yaitu:

1. *Capacitated VRP*, merupakan kelas VRP yang paling sederhana dan yang paling banyak dipelajari dimana kendala yang ada hanya berupa kapasitas kendaraan yang terbatas.
2. *Distance Constrained VRP*, merupakan VRP dengan kendala batasan panjang rute.
3. *VRP with Time Windows*, yaitu kasus VRP dimana setiap konsumen memiliki batasan rentang waktu pelayanan.
4. *VRP with Pickup and Delivery*, merupakan VRP dengan pelayanan campuran, yaitu pengiriman dan pengambilan barang dalam satu rute.

5. *VRP with Backhauls*, dimana pengambilan baru dapat dilakukan setelah semua pengiriman selesai.

Seiring perkembangan permasalahan distribusi menyebabkan timbulnya jenis-jenis baru dalam permasalahan VRP. Terdapat batasan atau kendala yang digabungkan untuk menyelesaikan permasalahan agar sesuai dengan realita di dunia nyata. Suprayogi (2003), telah mengembangkan beberapa varian dari VRP seperti:

1. *VRP with Heterogeneous Fleet*, dimana pada VRP ini memiliki karakteristik berupa kendaraan yang digunakan memiliki kapasitas yang berbeda dan telah diketahui dengan pasti jumlah dan jenisnya.
2. *VRP with Time Windows*, dimana VRP memiliki pelanggan maupun depot mempunyai batas waktu pelayanan, yaitu terdapat rentang waktu di mana hanya pada waktu tersebut pelayanan dapat dilakukan.
3. *VRP with Pickup and Delivery*, karakteristik dari VRP ini yaitu kendaraan dapat melakukan pengantaran dan pengambilan barang sekaligus.
4. *VRP with Split delivery*, VRP yang memungkinkan setiap pelanggan dapat dilayani oleh lebih dari satu kendaraan.
5. *VRP with Multiple Trip*, karakteristik dari VRP ini memungkinkan setiap kendaraan dapat memiliki lebih dari satu rute perjalanan dalam satu horizon perencanaan namun harus kembali ke depot terlebih dahulu sebelum mengunjungi pelanggan pada rute selanjutnya.
6. *VRP with Multiple Depot*, berarti perusahaan dapat memiliki lebih dari satu depot, sehingga dapat terjadi satu kendaraan berangkat dari satu depot dan mengakhiri rute pada depot yang lain.
7. *VRP with Multiple Product and Compartment*, pada VRP ini pelanggan dapat memesan lebih dari satu jenis barang dan masing-masing barang tersebut diletakkan pada kompartemen yang berbeda.

8. *Stochastic VRP*, merupakan VRP yang memiliki karakteristik berupa permintaan pelanggan yang tidak diketahui dengan pasti, waktu perjalanan yang berubah-ubah, dan setiap pelanggan mungkin tidak dikunjungi setiap hari.
9. *Dynamic VRP*, merupakan VRP yang memungkinkan terdapat sejumlah pelanggan yang mendapatkan pelayanan yang berbeda-beda setiap waktunya.
10. *Periodic VRP*, merupakan VRP yang hari kunjungan ke pelanggan ditetapkan dalam jangka waktu tertentu secara berkala atau periodic.

2.3 Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Multiple Products

Salah satu varian dari VRP yaitu *VRP with Heterogeneous* dimana kendaraan yang bertugas mengirim barang memiliki karakteristik yang berbeda seperti kapasitas angkut, kecepatan dan biaya. Menurut Choi dan Tcha (2007), ada tiga versi dari *VRP with Heterogeneous* yang telah dipelajari, pertama kali diperkenalkan oleh Golden, dkk. (1984), dimana biaya variabel adalah *uniform* untuk semua tipe kendaraan dengan jumlah kendaraan yang tersedia diasumsikan tidak terbatas. Versi ini disebut *Fleet Size and Mix VRP*, Golden, dkk. (1984) atau *Fleet size and Composition VRP*, Gheysen, dkk. (1986). Versi yang kedua mempertimbangkan biaya variabel yang tergantung tipe kendaraan, yang berbeda dengan versi pertama yang tidak mempertimbangkannya. Versi ini disebut dengan *Heterogeneous Fleet VRP*, Gendreau, dkk. (1999), *VFM with variable unit running cost*, Salhi dan Rand, (1992), atau *Fleet Mix VRP*, Wassan dan Osman (2002). Versi yang terakhir disebut dengan *VRP with Heterogeneous Fleet of Vehicles*, Taillard (1999), atau *The Heterogeneous Fixed Fleet VRP*, Tarantilis, dkk. (2004) yang merupakan generalisasi dari versi yang kedua dengan jumlah kendaraan untuk masing-masing tipe terbatas.

Pada perkembangan *VRP with Time Windows*, menurut Kang, dkk. (2008) menjelaskan bahwa *VRPTW* dapat diklasifikasikan ke dalam *VRP with Hard Time Windows* dan *VRP with Soft Time Windows*. Pada *VRP with Hard Time*

Windows pelanggan hanya dapat dilayani selama *time windows* yang sudah ditentukan. Sedangkan pada *VRP with Soft Time Windows* mereka dapat dilayani kapan saja, namun akan mendapat hukuman jika diluar *time windows*. Sedangkan untuk *VRP with Multiple Product*, menurut Amorim, dkk (2014) pada dasarnya produk dapat dibagi menjadi tiga kategori: kering, dingin (segar) dan beku. Oleh karena itu biasanya kendaraan memiliki kompartemen yang dapat diatur ke suhu yang berbeda saat digunakan.

Heterogenous Fleet Vehicle Routing Problem with Time Windows and Multiple Products merupakan varian VRP yang memiliki 3 batasan sekaligus, yaitu kendaraan yang digunakan untuk melayani pelanggan memiliki karakteristik yang berbeda, setiap pelanggan memiliki batasan rentang waktu pelayanan, dan produk yang dikirim ke pelanggan memiliki varian yang berbeda. Terdapat penelitian yang terkait dengan HFVRPTWMP yaitu penelitian Amador, dkk (2014) menggunakan metode *Bacterial Meta Heuristic*, penelitian Anggela (2017), dengan menggunakan Algoritma *Sequential Insertion*, penelitian De La Cruz, dkk. (2013) dengan menggunakan metode *ant colony system* dan *tabu search*, dan penelitian Desiana (2016), menggunakan algoritma genetika yang dapat menyelesaikan permasalahan keterlambatan dan meminimalkan biaya transportasi sebesar 5,83%. Pada penelitiannya digunakan ukuran populasi sebanyak 30, probabilitas kawin silang 0,94, dan probabilitas mutasi 0,01. Sedangkan untuk seleksi menggunakan roda rolet, serta penyilangan dan mutasi menggunakan berbasis urutan. Penelitian De La Cruz, dkk. (2013) dengan menggunakan metode *ant colony system* dan *tabu search* dapat meminimalkan jarak sebesar 5,7%.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	VRP			Metode
		Time Windows	Heterogeneous Fleet	Multiple Products	
1	Choi dan Tcha (2007)		✓		Pendekatan Column Generation berbasis LP
2	Kang, dkk. (2008)	✓			Tabu Search
3	De La Cruz, dkk. (2013)	✓	✓	✓	Ant Colony System
4	Amador, dkk (2014)	✓	✓	✓	Bacterial Meta Heuristic
5	Anggela (2017)	✓	✓	✓	Algoritma <i>Sequential Insertion</i>
6	Desiana (2016)	✓	✓	✓	Algoritma Genetika
7	Penelitian ini	✓	✓	✓	Algoritma Genetika

2.4 Model Matematis HFVRPTWMP

Model matematis yang digunakan pada permasalahan HFVRPWTMP adalah sebagai berikut (Tan, dkk.2001):

Indeks:

i : indeks *customer*

j : indeks *customer*

dimana $i \neq j$

k : indeks kendaraan

Parameter:

K : total jumlah kendaraan

N : total jumlah *customer*

c_{ijk} : biaya yang dikeluarkan dari node i ke j dengan kendaraan k

t_{ij} : waktu tempuh anatar node i ke j

m_i : permintaan node i

q_k : kapasitas kendaraan k

e_i : waktu kedataan awal

l_i : waktu kedatangan akhir

f_i : waktu pelayanan

r_k : waktu maksimum rute yang diijinkan

t_i : waktu kedatangan

w_i : waktu tunggu

Variabel keputusan

$x_{ijk} = 1$ jika dari i ke j menggunakan kendaraan k

$x_{ijk} = 0$ jika lainnya

$$\sum_{i=0}^N \sum_{j=0, j \neq i}^N \sum_{k=1}^K c_{ijk} \cdot x_{ijk} \dots\dots\dots(1)$$

$$\sum_{k=1}^K \sum_{j=1}^N x_{ijk} \leq K \text{ for } i = 0 \dots\dots\dots(2)$$

$$\sum_{j=1}^N x_{ijk} = \sum_{j=1}^N x_{jik} \leq 1 \text{ for } i = 0 \text{ and } k \in \{1, \dots, K\} \dots\dots\dots(3)$$

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^N x_{ijk} = 1 \text{ for } i \in \{1, \dots, N\} \dots\dots\dots(4)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^N x_{ijk} = 1 \text{ for } j \in \{1, \dots, N\} \dots\dots\dots(5)$$

$$\sum_{i=1}^N m_i \sum_{j=0, j \neq i}^N X_{ijk} \leq q_k \text{ for } k \in \{1, \dots, K\} \dots \dots \dots (6)$$

$$\sum_{i=0}^N \sum_{j=0, j \neq i}^N x_{ijk} (t_{ij} + f_i + w_i) \leq r_k \text{ for } k \in \{1, \dots, K\} \dots \dots \dots (7)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^N x_{ijk} (t_i + t_{ij} + f_i + w_i) \leq t_j \text{ for } j \in \{1, \dots, N\} \dots \dots \dots (8)$$

$$t_0 = w_0 = f_0 = 0 \dots \dots \dots (9)$$

$$e_i \leq (t_i + w_i) \leq l_i \text{ for } i \in \{1, \dots, N\} \dots \dots \dots (10)$$

$$x_{ijk} \in \{1, 0\} \dots \dots \dots (11)$$

Formula (1) adalah fungsi tujuan dari masalah. Kendala (2) menentukan rute K yang keluar dari depot. Persamaan (3) memastikan setiap rute berawal dan berakhir di depot pusat. Persamaan (4) dan (5) mendefinisikan bahwa setiap node pelanggan dapat dikunjungi hanya sekali satu kendaraan. Persamaan (6) adalah keterbatasan kapasitas. Persamaan (7) adalah kendala waktu perjalanan maksimal. Pembatas (8) sampai (10) menentukan jendela waktu.

2.5 Algoritma Genetika

Algoritma genetika pada hakikatnya diciptakan oleh satu orang yaitu John Holland pada tahun 1960an (Coley, 1999). Algoritma genetika merupakan suatu metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah Teori Evolusi Darwin, yaitu suatu individu tercipta secara acak kemudian berkembang biak melalui proses reproduksi sehingga terbentuk sekumpulan individu sebagai suatu populasi (Zukhri, 2014). Ada beberapa definisi penting yang perlu diperhatikan dalam penyelesaian permasalahan dengan algoritma genetika seperti (Satriyanto, 2009):

1. Gen adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan. Gen ini bisa berupa nilai biner, float, integer maupun karakter atau kombinatorial.
2. Allele adalah nilai dari gen.
3. Kromosom adalah gabungan dari gen-gen yang membentuk nilai tertentu.
4. Individu adalah menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.

5. Populasi adalah sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. Generasi adalah menyatakan satu siklus proses evolusi atau satu itersi di dalam algoritma genetika.

Pada algoritma genetika terdapat beberapa langkah dalam penyelesaiannya. Menurut Haupt dan Haupt (2004), struktur dasar algoritma genetika terdiri atas beberapa langkah yaitu:

1. Inisialisasi populasi.
2. Evaluasi populasi.
3. Seleksi populasi yang akan dikenai operator genetika.
4. Proses penyilangan pasangan kromosom tertentu.
5. Evaluasi populasi baru.
6. Ulangi dari langkah 3 selama syarat berhenti belum terpenuhi.

Terdapat beberapa kelebihan algoritma genetika dibandingkan dengan algoritma yang lain. Menurut Gen dan Cheng (1997), kelebihan-kelebihan tersebut adalah:

1. Algoritma ini hanya melakukan sedikit perhitungan matematis yang berhubungan dengan masalah yang ingin diselesaikan.
2. Operator-operator evolusi membuat algoritma ini sangat efektif pada pencarian global.
3. Algoritma ini memiliki fleksibilitas yang tinggi untuk dihibridkan dengan metode pencarian lain supaya lebih efektif.

Sedangkan menurut Widodo (2012), ada tiga keuntungan utama dalam mengaplikasikan algoritma genetika pada masalah-masalah optimasi seperti:

1. Algoritma genetika tidak memerlukan kebutuhan matematis banyak mengenai masalah optimasi.
2. Algoritma genetika menyediakan banyak fleksibilitas untuk digabungkan dengan metode heuristik yang tergantung dominan, untuk membuat implementasi yang efisien pada masalah-masalah khusus.
3. Kemudahan dan kenyamanan pada operator-operator evolusi membuat algoritma genetika sangat efektif dalam melakukan pencarian global

2.6 Komponen-Komponen Algoritma Genetika

Menurut Kusumadewi (2003), ada enam komponen utama dalam algoritma genetika yaitu:

1. Teknik Penyandian

Pada Algoritma genetika inisialisasi merupakan hal yang penting karena peran kromosom sebagai representasi dari sebuah solusi permasalahan. Pada prinsipnya, tidak ada aturan khusus mengenai pengkodean ini, kromosom dapat dirancang dengan kode-kode tertentu dengan persyaratan dapat diproses oleh operator-operator algoritma genetika dan merupakan representasi penyelesaian masalah yang akan dioptimasi (Zukhri, 2014). Kromosom dapat dipresentasikan dengan menggunakan (Kusumadewi, 2003):

- a. String bit : 10011, 01101, 11101, dst.
- b. Bilangan riil : 65.65, -67.98, 562,88, dst.
- c. Elemen permutasi : E2, E10, E5, dst.
- d. Daftar aturan : R1, R2, R3, dst.
- e. Elemen program : pemrograman genetika
- f. Struktur lain.

Pada penelitian Siregar (2012), kromosom dipresentasikan dengan *integer representation chromosome* ialah dimana deretan bilangan dalam kromosom merepresentasikan kendaraan mana yang akan melewati titik tersebut, dengan posisi kromosom merepresentasikan titik yang akan dilewati. Sebagai contoh dapat dilihat pada gambar 2.1.

1	2	3	4	5	6	7	8	21
4	6	4	5	1	3	3	1	6

Gambar 2.1 Contoh representasi kromosom integer

Rute yang ditempuh oleh kendaraan 4 di mulai dari depot kemudian menuju titik 1 lalu ke titik 3, kemudian kembali ke depot.

2. Pembentukan Populasi Awal

Pembangkitan populasi awal dilakukan secara acak atau bisa dilakukan melalui proses tertentu. Ada beberapa teknik dalam pembangkitan populasi awal diantaranya (Satriyanto,2009)::

a. Random generator

Inti dari cara ini adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan.

b. Pendekatan tertentu

Cara ini adalah dengan memasukkan nilai tertentu ke dalam gen dari populasi awal yang dibentuk.

c. Permutasi gen

Salah satu cara permutasi gen dalam pembangkitan populasi awal adalah penggunaan permutasi Josephus.

3. Evaluasi

Ada 2 hal yang harus dilakukan dalam evaluasi kromosom, yaitu: evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness* (Kusumadewi, 2003). Kromosom-kromosom akan dievaluasi menggunakan fungsi *fitness*, dimana fungsi *fitness* digunakan untuk mengukur seberapa baik dari solusi tersebut. Pada permasalahan VRP terdapat kendala-kendala yang tidak boleh dilanggar, maka terdapat fungsi penalti apabila kendala-kendala tersebut dilanggar. Terdapat rumus fungsi fitness dengan pinalti (Zuhkri, 2014):

$$eval(v) = f(v) \cdot p(v)$$

$$p(v) = 1, \text{ untuk kromosom layak}$$

$$p(v) > 1, \text{ untuk kromosom tidak layak}$$

Dimana $eval(v)$ adalah fungsi *fitness*, $f(v)$ adalah fungsi tujuan, dan $p(v)$ adalah fungsi pinalti.

4. Seleksi

Setelah kromosom-kromosom dievaluasi dengan nilai fitness maka akan dilakukan seleksi untuk menentukan kromosom mana yang

memiliki kualitas bagus untuk menjadi induk. Terdapat beberapa metode seleksi dari induk, antara lain (Kusumadewi, 2003):

- a. *Rank-based fitness assignment.*
- b. *Roulette wheel selection.*
- c. *Stochastic universal sampling.*
- d. *Local selection.*
- e. *Truncation selection*
- f. *Tournament selection.*

Salah satu metode seleksi yang sering digunakan adalah seleksi roda rolet (*Roulette wheel selection*). Adapun langkah-langkah yang dilakukan pada seleksi roda rolet (Zukhri, 2014) yaitu:

- a. Membangkitkan bilangan random r yang bernilai anrata 0 sampai 1 ($0 \leq r \leq 1$).
- b. Jika $r < \frac{eval(v_1)}{\sum_{j=1}^{ukuran\ populasi} eval(v_j)}$, maka pilih kromosom pertama
atau jika $\frac{\sum_{j=1}^{i-1} eval(v_j)}{\sum_{j=1}^{ukuran\ populasi} eval(v)} \leq r < \frac{\sum_{j=1}^i eval(v_j)}{\sum_{j=1}^{ukuran\ populasi} eval(v)}$,
maka pilih kromosom ke- i .
- c. Ulangi kedua langkah di atas sebanyak kromosom dalam sebuah populasi.

5. Penyilangan dan Mutasi

Penyilangan dilakukan untuk mendapatkan kombinasi yang lebih baik antara satu kromosom dengan kromosom yang lain. Penyilangan dimulai dari menentukan kromosom yang terpilih dari proses seleksi yang selanjutnya akan dibangkitkan bilangan random sejumlah kromosom dalam populasi. Apabila bilangan random kromosom tersebut kurang dari probablitas penyilangan (P_c), maka kromosom tersebut terpilih sebagai induk. Hartono (2011), menyebutkan beberapa teknik pada proses penyilangan sebagai berikut:

a. *One point crossover*

Sebuah titik dipilih , selanjutnya *string* biner mulai dari awal kromosom sampai sampai dengan titik tersebut disalin dari salah satu orang tua ke keturunannya, kemudian sisa bit keturunan disalin dari orang tua yang kedua.

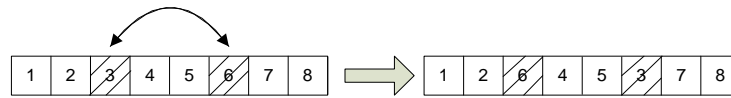
b. *Two point crossover*

Dua titik *crossover* dipilih, selanjutnya *string* biner mulai dari awal kromosom sampai dengan titik *crossover* pertama disalin dari salah satu orangtua ke keturunannya kemudian mulai dari titik *crossover* pertama sampai dengan titik kedua disalin dari orang tua kedua. Sisanya disalin dari orangtua pertama.

Mutasi bertujuan untuk mengubah gen-gen tertentu dari kromosom. Kromosom yang masuk proses mutasi dipilih berdasarkan mekanisme random. Bilangan random yang dibangkitkan sejumlah banyaknya kromosom dalam satu populasi. Apabila bilangan random pada kromosom kurang dari samadengan probabilitas mutasi, maka kromosom tersebut terpilih untuk melakukan proses mutasi. Kromosom yang dimutasi adalah kromosom anak hasil dari penyilangan. Terdapat beberapa teknik dalam proses mutase seperti (Zukhri,2014):

- a. Mutasi berbasis posisi
- b. Mutasi berbasis urutan
- c. Mutasi berbasis campur aduk

Salah satu metode yang biasa digunakan adalah mutasi berbasis urutan dimana dilakukan pemilihan dua gen secara acak, kemudian kedua gen tersebut akan ditukar posisinya. Sehingga urutan kedua-dua gen sesudah mutasi menjadi kebalikan dari urutannya sebelum mutasi. Pemilihan dua gen tersebut dengan cara membangkitkan dua bilangan acak yang ada pada kromosom. Sebagai contoh bisa dilihat pada gambar 2.2 (Zukhri,2014).



Gambar 2.2 Ilustrasi mutasi berbasis urutan

6. Penentuan Parameter

Menurut Obitko (1998), terdapat tiga parameter didalam algoritma genetika yang menjadi standar yaitu probabilitas penyilangan (P_c), probabilitas mutasi (P_M), dan ukuran populasi (pop_size). Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan. Ada beberapa rekomendasi yang bisa digunakan, antara lain (Kusumadewi, 2003):

1. Untuk permasalahan yang memiliki kawasan solusi cukup besar, De Jong merekomendasikan untuk nilai parameter control: $(pop_size; P_M; P_c) = (50; 0,6; 0,001)$.
2. Bila rata-rata fitness setiap generasi digunakan sebagai indikator, maka grefenstette merekomendasikan: $(pop_size; P_M; P_c) = (30; 0,95; 0,01)$.
3. Bila fitness dari individu terbaik dipantau pada tiap generasi, maka usulannya adalah: $(pop_size; P_M; P_c) = (80; 0,45; 0,01)$.
4. Ukuran populasi sebaiknya tidak lebih kecil dari 30, untuk sembarang jenis permasalahan.

2.7 Update Generasi dan Syarat Berhenti

Setelah terbentuk populasi baru, populasi tersebut dipakai untuk menggantikan populasi lama. Di dalam algoritma genetika suatu populasi dirancang untuk memiliki jumlah individu yang selalu tetap pada setiap generasi. Dalam proses *update* kromosom dengan *fitness* yang terbaik selalu dipertahankan dari generasi ke generasi (Haupt dan Haupt, 2004). Menurut Arifudin (2016), terdapat dua model *update* yang bisa digunakan yaitu:

1. Model *generational*

Dalam algoritma berevolusi dikenal dengan skema penggantian populasi yang disebut *generational replacement* yang berarti semua individu dari suatu generasi digantikan sekaligus oleh jumlah individu baru hasil pindah silang dan mutasi (Munawir, dkk. 2016).

2. Model *steady state*

Pada model ini tidak semua kromosom digantikan hanya pada sejumlah kromosom tertentu. Prosedur pergantian dengan *steady state* yaitu (Munawir, dkk. 2016):

- a. Mengganti individu yang memiliki nilai *fitness* terkecil.
- b. Mengganti individu yang paling tua.
- c. Membangkitkan anak dengan kedua org tua, apabila anak memiliki nilai *fitness* lebih tertinggi, maka anak bisa menggantikan orang tua yang memiliki nilai *fitness* rendah.

Proses optimasi yang dilakukan dengan algoritma genetika akan berhenti setelah syarat berhenti terpenuhi (Ding dan Gasvoda, 2004). Menurut Khotimah (2005) ada beberapa criteria berhenti yang sering digunakan antara lain:

1. Berhenti pada generasi tertentu.
2. Berhenti setelah dalam beberapa generasi berturut-turut didapat nilai *fitness* terbaik tidak berubah.
3. Berhenti bila n generasi selanjutnya tidak didapat nilai *fitness* yang lebih tinggi.